

DETAILED ACTION

Response to Arguments

Response: Arguments / Expert Opinion

1. Applicants argue:

- 1.1 Applicants primarily rely on opinion / declaration under 37 C.F.R. 1.132.
- 1.2 Section 6 of the Declaration begins to attempt to demonstrate the differences between the cited art and the claimed invention.
- 1.3 Section 6 of the Declaration recites one meaning for the phrase "Computer Virtualization", which states that it means "hiding various kinds of physical computing resources from the user".
- 1.4 Section 7 lists a number of advantages of using VMware software for its user base. The section also concludes that the "the very authors of the VMware User Manual cited by the Examiner do not regard their own product as a possible solution of the types of computing configurations described and claimed in the Application" from there not being an explicit listing, on the specifically cited page (history.html), directed to High-Performance Computing, parallel computing, or cluster computing.

2. Examiner Response:

- 2.1 Regarding subsection 3 *supra*, computer virtualization is a very broad subject and has many meaning. The subscription to the meaning provided by Applicants would unduly limit the field.
- 2.2 Regarding subsection 4 *supra*, the reached conclusion does not appear to have a valid basis. Specifically, it is unknown how the Declaration could reach such a drastic and bold conclusion that the authors did not regard possible solutions even though 1) the declaration was not authored by all , or apparently any of the authors of VMware and 2) the sole basis for such assertion is based on a single webpage showing the history of virtualization and its basics.

3. Applicants argue:

- 3.1 In section 8, the Declaration states "other virtualization technologies, such as VMware [] partition a single computer into a number of smaller, weaker sub-systems".

Art Unit: 2128

3.2 In section 9, Declaration asserts that "it is beyond the capabilities of [High-Performing Computing systems of the 1990's [] to turn the many computers into a single virtual computer, with a shared memory and a single operating system.

3.3 In section 9, Declaration further asserts "Such distributed computing techniques enable many processors to cooperate together in order to execute a single job."

4. **Examiner Response:**

4.1 Regarding subsection 1 *supra*, virtualization does not inherently partition a single computer into a number of smaller systems. In fact, virtualization can have a one-to-one correlation.

4.2 Regarding subsection 1 and 2 *supra*, the basis of this conclusion is not based on any submitted evidence, but is a matter of opinion. In fact, the conclusion reached in section 9 appears to conflict with the initial statements of that same section. Specifically, the concept of to "use standard computers and standard Ethernet interconnect between them" implies the aggregation of computer resources. This was done, as declared, "in order to perform distributed computing applications".

4.3 Regarding subsection 3 *supra*, indeed, that was the provided rejection. Clearly, virtualization has various advantages. This is not argued by the Declaration. Aggregation of computing resources in order to perform a single task is likewise admitted to by the Declaration. It is not unreasonable that, and in fact was concluded in the Office Action that, one of ordinary skill in the art, at the time of Applicants' invention, could and would want to combine the advantages having a virtualized environment with aggregation of computing resources. This would yield much more powerful virtualization system with various advantages of virtualization.

5. **Applicants argue:**

5.1 In section 10, Declaration states an opinion as to what is believed to be claimed.

5.2 In section 11, an opinion is presented on the basis that memory is shared amongst the distributed computing system in a manner which is "diametrically opposed to the shared memory structure that is needed to support a shared virtual machine".

Art Unit: 2128

6. Examiner Response:

6.1 Regarding subsection 2 *supra*, the statement seems to be conflicting with the previous Declaration.

Specifically, the resources disclosed in Okamoto are aggregated in order to perform a single job. It is not unreasonable that such a job be the execution of a VMware virtualization software. It is unclear how the conclusion of diametric opposition was reached. Nothing in the combination teaches away from such an embodiment.

7. Examiner Response:

7.1 With respect to section 12 of the Declaration, the statements are not incorrect.

8. Applicants argue:

8.1 In section 13, a question is raised: "If there is no virtual machine [] in Okamoto, what use would Okamoto have for a virtual machine implementer, and what could have led a person of ordinary skill in the art to consider Okamoto's operating system to be anything other than an operating system?"

8.2 Section 13 also makes a conclusionary statement that "there is no relation between 'virtual memory' and 'virtual machine'".

9. Examiner Response:

9.1 Regarding subsection 1 *supra*, to answer the first part of the question, as previously mentioned and agreed-upon, virtual machines have various advantages over their real-counterparts. Including, sandboxing, abstraction, hiding of hardware, etc. Hence, all of these features would prompt one of ordinary skill in the art to use Okamoto with a virtual machine. To answer the second part of the question, it is irrelevant as to what the person of ordinary skill in the art would "consider Okamoto's operating system to be anything other than an operating system". The question is whether it performs the same functions as a virtual machine implementer. A virtual machine implementer, by definition implements a virtual machine. Patentable weight is not attached to the naming convention attached to the structural and functional elements. Specifically, naming the "operating system" as

Art Unit: 2128

"virtual implementer", "widget modifier", or another term would not be controlling of its function or structural implementation. The question is whether the operating system disclosed by Okamoto, in combination with VMware renders the claim as a whole obvious. Indeed, it would appear to do just that.

9.2 Regarding subsection 2 *supra*, this issue is moot as the rejection is not making a correlation of "virtual memory" to "virtual machine".

10. Applicants argue:

10.1 In section 14, Declaration is made speculatively regarding one of ordinary skill in the art and what he or she might have done.

11. Examiner Response:

11.1 Regarding subsection 1 *supra*, the statements provided in section 14 are merely speculative with respect to the person of ordinary skill in the art. The Declaration in fact makes statements contrary to the intention of Okamoto's disclosure. Specifically, Okamoto aggregates resources to perform a single job, as is the position of the Office Action, and previously declared (**Section 9**). It would be counterproductive and against the spirit of Okamoto to install a VMware Workstation on multiple computers and not take advantages of Okamoto's disclosure (the aggregation of resources). Therefore, it would not appear that "this person might install multiple VMware Workstations on multiple computers", but rather that the person of ordinary skill in the art would install a single VMware Workstation as that single "job" of the aggregated computing resources.

12. Examiner Response:

12.1 Statements in section 15 have been previously addressed. Further statements in section 15 conflict with section 16 of the Declaration. Specifically, section 15 declares "VMware Workstation is not just an application program." Whereas, section 16 declares "VMware Workstation itself is a single-computer program". If the emphasis was implicitly made on "single-computer program", then

Art Unit: 2128

it is seen that Okamoto discloses aggregation of resources to perform a single job, whatever that job or program may be.

- 12.2 Section 16 alleges that adapting VMware Workstation to run over multiple separate computers would be a difficult, if not impossible task. Again, this is merely speculation. Okamoto's system is specifically designed to allow for aggregation of resources to perform a single job. This takes care of the adaptation required to run VMware Workstation on multiple computer. Meaning, VMware Workstation does not have to be adapted, as Okamoto already provides the functionality.

13. Applicants argue:

- 13.1 On page 2 of the Remarks two bullet points are made.
- 13.2 First bullet makes the argument of division vs. aggregation of resources.
- 13.3 Second bullet makes the argument that Okamoto does not mention or hint of a virtual machine.

14. Examiner Response:

- 14.1 Regarding subsection 2 *supra*, as previously addressed, Okamoto aggregates resources.

Furthermore, VMware does not inherently involve or pertain a division of resources. Further, VMware does not require multiple virtual machines, nor splitting resources among them. Specifically, VMware workstation is fully capable, and described as running a single virtual machine.

- 14.2 Regarding subsection 2 *supra*, the mention of "virtual machine" is not required, as the rejection must be taken as a whole. Okamoto discloses aggregation of multiple-computer resources to perform a single job / single program. VMware teaches providing functionality that is capable of creating a single, or even going beyond, and creating multiple virtual computers. The statement that "Okamoto does not mention or even hint at this sort of (virtual machine) functionality, not would his operating system (OS) be capable of supporting it" is not agreed-with. Specifically, Okamoto does not place restrictions on the types of jobs that can be performed, or the types of computer programs that can be executed. Therefore, one could and would have seen the advantages of running a virtual machine on an aggregated computing environment, as is claimed.

15. Consideration was indeed given to the previously submitted secondary consideration but the Declaration was unpersuasive. Specifically, it pointed out similar issues as above, and has thus been addressed explicitly herewithin.
16. There are no, and none are pointed-out by the Declaration, issues that would prevent one from making the combination. Further, there are many motivations that would allow and encourage one of ordinary skill in the art to make the combination. The various advantages are security, sandboxing, computer resource aggregation (lots of boxes methodology), less expensive super-computers, etc. Okamoto discloses aggregation of resources to perform a single job / run a single computer program. VMware Workstation is a program that, when ran, creates one or more virtual machines. Nothing in Okamoto or VMware prevents the virtual machine from working on Okamoto's system. Accordingly, the operating systems of Okamoto would be implementing the virtual machine, in concert with the VMware Workstation.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

17. Claims 1-4, 9-16, 22-30, and 32-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Okamoto (**US 5829041**), in view of VMware Workstation "User's Manual" Version 3.2, ("VMware"). Okamoto discloses: 1. (currently amended) A method for executing a software application in a plurality of computers having respective hardware resources said hardware resources comprising a respective

memory and a respective I/O device (**Fig 3 with emphasis on Computer 1 and Computer 2, as well as the implied networking between the computers; col: 2 line: 8-15**),

wherein said computers include a first computer and a second computer that intercommunication over a network (**col: 2 line: 8-15**), comprising the steps of:

running at least a first virtual machine implementer and a second virtual machine implementer on said first computer and said second computer, using said respective memory, wherein said first and second virtual machine implementers run separately and independently of one another on said first and second computers, respectively (**col: 5 line: 66 to col: 6 line: 7; the OS functions as the virtual machine implementer**).

Okamoto however does not expressly disclose:

executing a virtual machine on said computers, wherein sharing said virtual machine is shared between said first virtual machine implementer and said second virtual machine implementer using said respective I/O device in each of said first computer and said second computer to intercommunicate between said first computer and said second computer, and a guest operating system runs over said shared virtual machine.

Although Okamoto discloses executing a program executing transparently on the distributed system (**col: 5 line: 4-14**), and functionally said collection of computers would function and the individual VM implementers (Oss) would function as a virtual machine, in order to make the record abundantly clear, VMware is used to show that such a program as disclosed by Okamoto could be VMware's Workstation (Virtual Machine), and such program would be distributed among the computers in the fashion as disclosed by Okamoto, also that such program would execute a virtual machine with a guest operating system.

Specifically, VMware discloses executing a program which creates a virtual machine and contains a guest operating system. The combination of the two references would yield predictable results (**VMWare: page 21**).

It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to achieve the various goals of virtual computing, such as, for example, sandboxing and security afforded to virtual environments. Further motivations include crash-resistance and crash-recovery, which would benefit from the VM environment. In fact, Okamoto provides clear motivation showing that a single program (VMware's Virtual Machine, for example), creates "easier" **(col: 5 line: 36-42).**

The combination teaches: 2. (previously presented) The method according to claim 1, further comprising step of running said software application over said guest operating system, so that commands invoked by said software application are monitored or emulated by said first virtual machine implementer and said second virtual machine implementer on said first computer and said second computer, while said hardware resources of said first computer and said second computer are shared by communication over said network **(VMWare: page 21; this is inherent as the commands executed would be loaded into a single aggregated virtual space, and then properly executed with the respective computer and resource. This is evidenced by Okamoto: col: 5 line: 66 to col: 6 line: 7).**

The combination teaches: 3. (original) The method according to claim i, wherein at least one of said first virtual machine implementer and said second virtual machine implementer is a virtual machine monitor **(Okamoto: col: 5 line: 66 to col: 6 line: 7; this is the function of Okamoto's OS).**

The combination teaches: 4. (original) The method according to claim i, wherein at least one of said first virtual machine implementer and said second virtual machine implementer is an emulator **(Okamoto: col: 5 line: 66 to col: 6 line: 7 Okamoto's OS emulates a memory space that does not exist).**

9. (previously presented) The method according to claim i, further comprising the steps of:

providing a management system for said first virtual machine implementer and said second virtual machine implementer to control said first computer and said second computer, respectively, wherein said management system comprises a wrapper for receiving calls to a device driver from said first virtual machine implementer, said wrapper invoking said device driver according to a requirement of said first virtual machine implementer **(col: 16 line: 52-62: the OS functions as the management**

and the VMI, which must use device drivers to communicate with the memory of the first and second computer, and consequently functions as a wrapper for calls made to the virtual memory space, and direct them to the actual device drivers and actual memory space).

VMware discloses: 10. (previously presented) The method according to claim 9, further comprising the step of providing a virtual PCI controller for said management system to control a physical PCI controller in one of said computers **(VMware page 206; 247 - bottom).**

VMware discloses: 11. (previously presented) The method according to claim 9, further comprising the step of providing a virtual DMA controller for said management system to control a physical DMA controller in one of said computers **(VMware page 149).**

VMware discloses: 12. (previously presented) The method according to claim 11, further comprising the steps of:

providing a virtual PCI controller to control a physical PCI controller in one of said computers; and during a bootstrap phase of operation scanning a device list with said virtual PCI controller to identify devices having on-board DMA controllers **(VMware page page 149-150 top).**

Okamoto discloses: 13. (previously presented) The method according to claim 1, further comprising the steps of:

with said first virtual machine implementer and said second virtual machine implementer maintaining mirrors of a portion of said respective memory that is used by said guest operating system in each of said computers; write-invalidating at least a portion of a page of said respective memory in one of said computers; and transferring a valid copy of said portion of said page to said one computer from another of said computers via said network **(Okamoto: col: 11 line: 59 to col: 12 line: 10).**

As per claims 14-16, 22-30, 32-45, note the rejection of claims 1-4, 9-13 above. The Instant Claims recite substantially same limitations as the above-rejected claims and are therefore rejected under same prior-art teachings.

Art Unit: 2128

18. Claims 5-8, and 17-21, 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Okamoto (**US 5829041**), in view of VMware Workstation "User's Manual" Version 3.2, ("VMware") in view of Altman (**US 20040054517**).

As per claim 5, the combination of Okamoto and VMware fully disclose claim 1. The combination however does not expressly disclose: at least said first computer comprises a first virtual node comprising a first physical CPU of said first computer and a second virtual node comprising a second physical CPU of said first computer.

Altman however discloses said feature (**para 0046**).

It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to support multiprocessor environments, which are typically faster and cost less as technology advances. Using multiple cores / multiple processors allows for more compact machines which allows for more space to be used for additional computers, or other necessary features (such as cooling fixtures).

As per claims 6-8, note the rejection of claim 5 above. The Instant Claims recite substantially same limitations as the above-rejected claim and are therefore rejected under same prior-art teachings.

As per claims 17-21, note the rejection of claims 5-8 above. The Instant Claims recite substantially same limitations as the above-rejected claims and are therefore rejected under same prior-art teachings.

As per claim 31, note the rejection of claim 5 above. The Instant Claim recites substantially same limitations as the above-rejected claim and is therefore rejected under same prior-art teachings.

Conclusion

19. All claims are rejected.

20. The Instant Application is not currently in condition for allowance.

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from

Art Unit: 2128

the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David Silver whose telephone number is (571) 272-8634. The examiner can normally be reached on Monday thru Friday, 10am to 6:30pm. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kamini Shah can be reached on 571-272-2279. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/David Silver/
Examiner, Art Unit 2128